Seminar-3

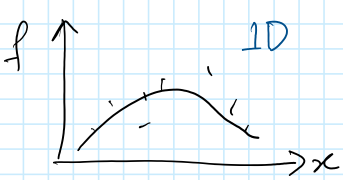
Thursday, September 18, 2025 3:27 PM

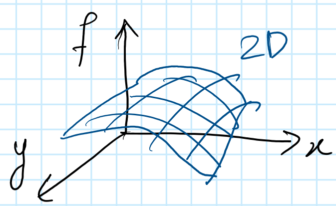
Last time: Given data (x_i, f_i), $i = 1, \dots, m$

Surrogate models obtained via:

→ Interpolation
- ⊕ Perfect at data points
- ⊖ bad for noisy data
- ⊖ Can be bad approx. between data points
  - ↳ Can fix with splines

→ Curve fitting (linear or nonlinear regression)
- ⊕ Good for noisy data
- ⊖ a guess is still a guess (type of function to use)
- ⊖ difficult in high dimensions (costly)





what happens if we have 100 dimensions?

Today: A different kind of surrogate model

Deep Neural Networks

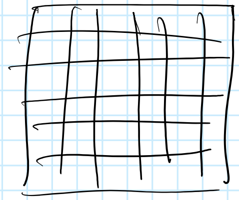
Give up having an explicit/simple parametric form of the model, but achieve good & efficient approximations in high dimensions.

example of problem in high dimensions:

10-15 years ago computers were good at things but not many things as well.

$\boxed{\text{cat}}$ or $\boxed{\text{dog}}$ → image identification

$\underline{|\text{cat}|}$ or $\underline{|\text{dog}|}$ $\rightarrow$ image identification



$\rightarrow$ each pixel is a dimension

1 M pixel $\rightarrow 10^6$ dimensions

Curve fitting is not a good choice!

<u>Idea ①:</u> Use simple functions to generate complex behavior (for example. by using function composition)

(Note: a linear function would not work, because

$$L_1(x) = ax + b$$

$$x \xrightarrow{L_1} L_1(x) \xrightarrow{L_1} L_1(L_1(x))$$

$$L_1(ax+b) = a^2x + ab + b$$

$\hookrightarrow$ still a linear function

can not create complex behavior.

<u>Idea ②:</u> Mimic how the brain works (how neurons interact)

a) Neurons have <u>active</u> & <u>inactive</u> <u>states</u>

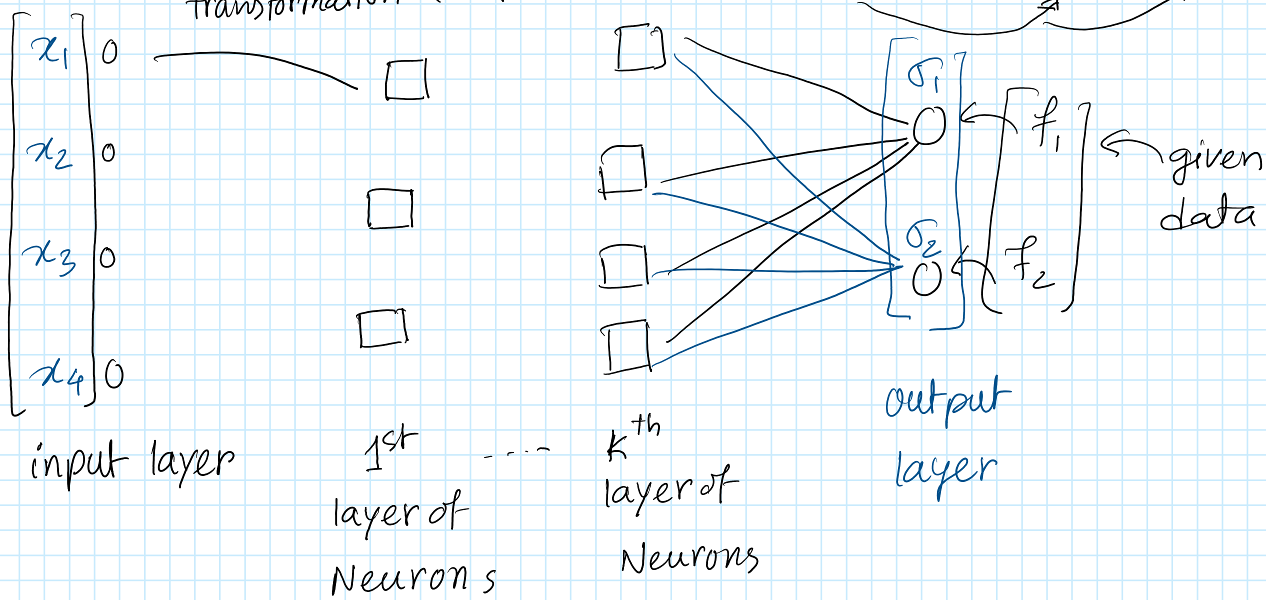
b) Neuronal connections: Signals are transferred from a bunch of neurons to another bunch and so on.

(layers) (layers)

c) Transmission of signals can be modified/biased and so on.

c) Transmission of signals can be modified/biased

transformation (depend on parameters like $\underline{a}x + \underline{b}$



__Goal is:__ Find the parameters in the neural network, such that we minimize the

$$\sum_{i=1}^{m} \| \underline{\sigma}_i - \underline{f}_i \|_2^2$$

# of given data points

Ⓠ Do we have to change the activation function between layers?

Ⓐ No, but we can change the parameters for the biases (a's, b's)

$$\min_{\substack{a's \\ b's}} \sum \| \qquad \|_2^2$$

→ we are trying to "learn" these to build the model

b's → we are trying to "learn" these to build the model
(find)

Examples of activation functions:

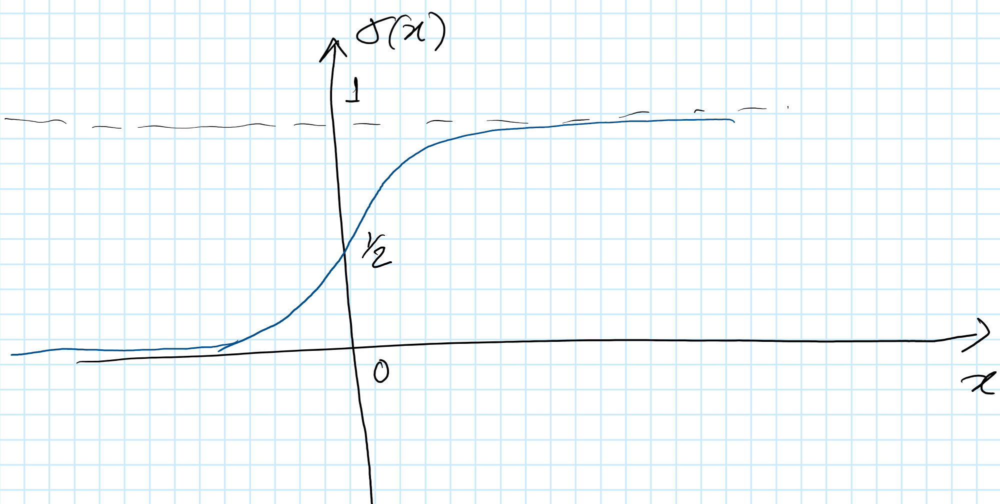
we want - zero (non-active)

- Non-zero (active)

The sigmoid function,

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\sigma'(x) = \sigma(x) \cdot (1-\sigma(x))$$



Q
How to affect the slope of the transition, and where it's value $= \frac{1}{2}$?

A by linear transformation, $\sigma(\alpha(x-\beta))$

→ produce the shift

→ affects how steep this is

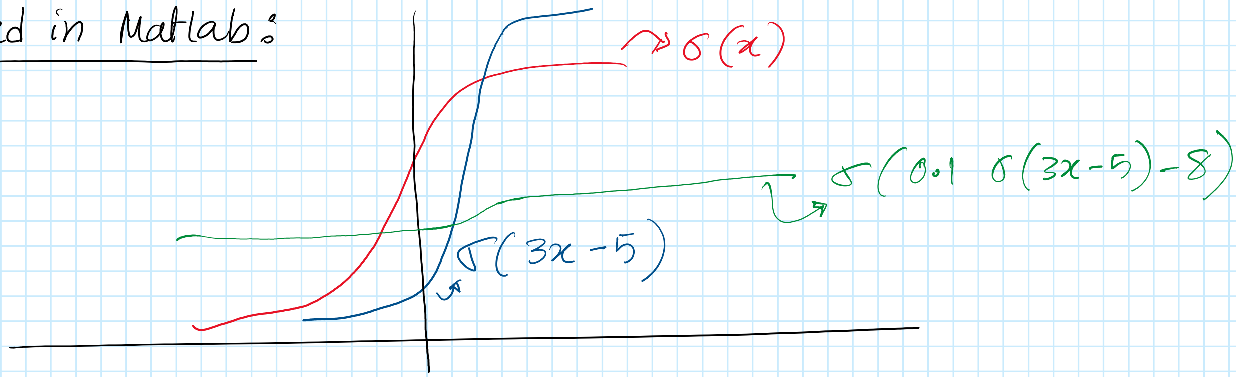
Note:

$$\sigma(a_2(\sigma(a_1 x + b_1)) + b_2) \neq (\sigma \circ \sigma) x$$

(not same)

(not same)

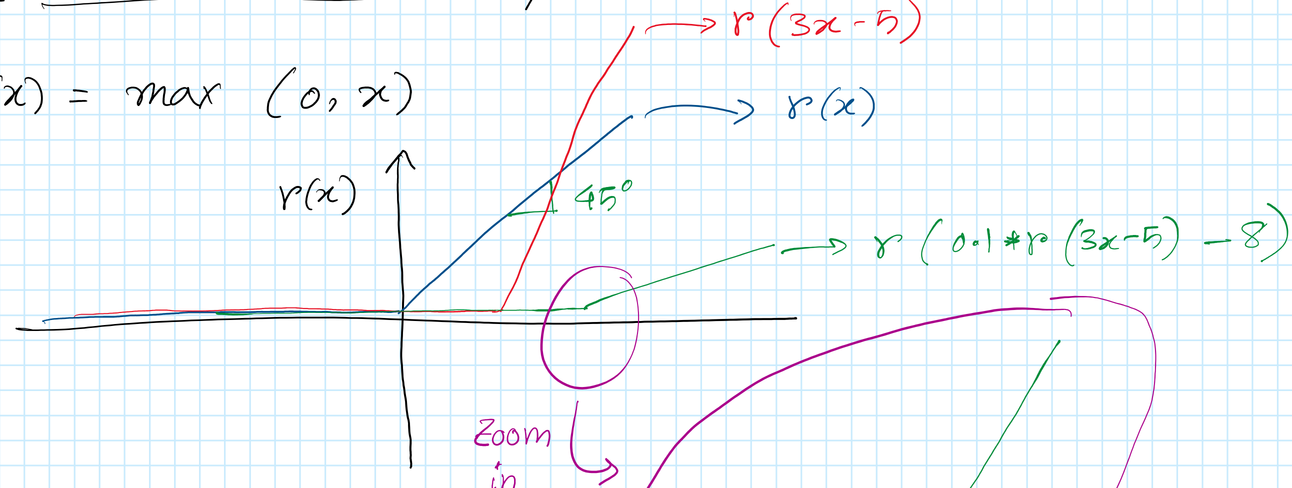
Showed in Matlab:

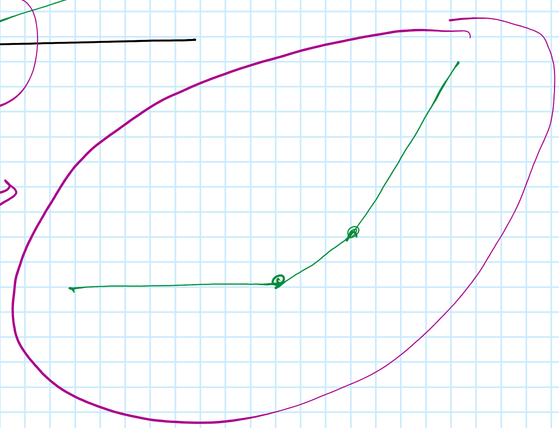


The ReLu

(Rectified Linear Unit)

$r(x) = \max(0, x)$





Similarities to previous models

↳ Solve a non-linear least squares minimization problem.
[recall: non-linear in unknowns (a's, b's)]
(use Gauss-Newton or Levenberg-Marquadt method)
①

Differences:

↳ Hard to tell how the function looks like.

Recall: The minimization problem

final layer

$$\min_{\substack{a's \\ b's}} \sum_{i=1}^{m} \| \underline{f}_i - \underline{\sigma}_i \|_2^2$$

↳ large dimensions, where methods in 1 are no longer efficient

↳ Need to use SGD (Stochastic Gradient Descent)